IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HARUMI ANNE KUNO	Examiner:	PANTOLIANO JR, R
Serial No.:	10/695,461	Group Art Unit:	2194
Filed:	October 28, 2003	Docket No.:	200207002-1 (HPCO.125PA)
Title:	METHOD AND APPARATUS FOR INTERFACING WITH A DISTRIBUTED COMPUTING SERVICE		

Board of Patent Appeals and Interferences
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Notice of Non-Compliant Appeal Brief mailed March 28th, 2008 this replacement Appeal Brief submitted pursuant to 37 C.F.R. § 41.37 for the above-referenced patent application. This Appeal Brief is intended to replace the Appeal Brief filed on March 4, 2008.

I. Real Party in Interest

The real party in interest is Hewlett-Packard Development Company, L.P., having a place of business Houston, Texas. The above referenced patent application is assigned to Hewlett-Packard Development Company, L.P.

II. Related Appeals and Interferences

Appellant is unaware of any related appeals, interferences or judicial proceedings.

III. Status of Claims

Claims 1-4, 6-16, 18-19, 21-23, and 25-28 are rejected and are presented for appeal. Claims 5, 17, 20, and 24 have been cancelled and withdrawn from consideration. The appealed claims are in the attached Appendix of Appealed Claims.

IV. Status of Amendments

No Amendment after Final Rejection was filed.

V. Summary of Claimed Subject Matter

In the embodiment set forth in claim 1, the invention provides a processor-implemented (FIG. 8, #802) method for interfacing with a distributed computing service ([0005], [0019]). The method comprises accessing (FIG. 6, #608; FIG. 7, #722) an ontology specification (FIG. 1, #118; FIG. 5, #512) describing messages of the distributed computing service. The method accesses (FIG. 6, #614; [0032], [0053]) a semantic interpretation specification (FIG. 1, #120; FIG. 5, #510) that describes rules for semantically handling the messages, as specified in the ontology specification, with the distributed computing service. The method further enters (FIG. 3, #304; FIG. 5, #508; [0032], [0039]) the semantic interpretation specification into a rules engine adapted for providing processor executable procedures (FIG. 5, 510; [0031], [0044]). A set of procedures is obtained from the rules engine for interacting with the distributed service based on the semantic interpretation specification (FIG. 3, #306; FIG. 5, #508, 510; [0032], [0039], [0044]). The method receives a request for interfacing with the distributed service (FIG 5, #504; [0044]; FIG. 7, #702; [0048]), and interfaces with the distributed computing service using the set of procedures in response to the request. The interfacing comprises forming distributed computing service messages based on the ontology specification (FIG. 5, #504, #506, #514; [0044]).

Claim 7 recites an apparatus comprising a data transfer interface for providing data connections to a distributed computing service (FIG. 1, #106, [0034]; FIG. 8, #828; [0054]). The apparatus also includes a processor (FIG. 8, #802; [0052]-[0053]) that is arranged to access (FIG. 6, #608; FIG. 7, #722) an ontology specification (FIG. 1, #118; FIG. 5, #512) describing messages of the distributed computing service. The processor also accesses (FIG. 6, #614; [0032], [0053]) a semantic interpretation specification (FIG. 1, #120; FIG. 5, #510) that describes rules for semantically handling the messages, as specified in the ontology specification, used to interface with the distributed computing service.

The semantic interpretation specification is entered (FIG. 3, #304; FIG. 5, #508; [0032], [0039]) by the processor into a rules engine adapted for providing processor executable procedures (FIG. 5, 510; [0031], [0044]). The processor obtains a set of procedures from the rules engine for interacting with the data

transfer service based on the semantic interpretation specification (FIG. 3, #306; FIG. 5, #508, 510; [0032], [0039], [0044]). The processor interfaces with the distributed computing service via the data transfer interface using the set of procedures, wherein the interfacing includes forming distributed computing service messages based on the ontology specification (FIG. 5, #504, #506, #514; [0044]).

In another embodiment as found in claim 13, a computer-readable medium (FIG. 8, #812, #814, #816, #818; [0055]) is configured with instructions for causing a processor to perform steps comprising accessing (FIG. 6, #608; FIG. 7, #722) an ontology specification (FIG. 1, #118; FIG. 5, #512) describing messages of the distributed computing service. The steps further include accessing (FIG. 6, #614; [0032], [0053]) a semantic interpretation specification (FIG. 1, #120; FIG. 5, #510) describing rules for semantically handling the messages, as specified in the ontology specification, with a distributed computing service. The semantic interpretation specification is entered (FIG. 3, #304; FIG. 5, #508; [0032], [0039]) into a rules engine adapted for providing processor executable procedures (FIG. 5, 510; [0031], [0044]). The step includes obtaining a set of procedures from the rules engine for interacting with the distributed service based on the semantic interpretation specification (FIG. 3, #306; FIG. 5, #508, 510; [0032], [0039], [0044]). A request is received for interfacing with the distributed service (FIG 5, #504; [0044]; FIG. 7, #702; [0048]), and the steps include interfacing with the distributed computing service using the set of procedures in response to the request, wherein the interfacing comprises forming distributed computing service messages based on the ontology specification (FIG. 5, #504, #506, #514; [0044]).

According to claim 19, a system comprises means (FIG. 8, #800) for providing a distributed computing service ([0005], [0019]). Means are provided for storing (FIG. 1, #108; FIG. 8, #824, #826; [0035], [0054]) an ontology specification (FIG. 1, #118; FIG. 5, #512) describing messages of the distributed computing service. Further means are provided for storing (FIG. 1, #108; FIG. 8, #824, #826; [0035], [0054]) a semantic interpretation specification (FIG. 1, #120; FIG. 5, #510) that describes rules for semantically handling the messages, as specified in the ontology specification, used to interface with the distributed computing service. The system further includes means for accessing (FIG. 6,

#614; [0032], [0053]; FIG. 8, #800; [0052]) the semantic interpretation specification for entry (FIG. 3, #304; FIG. 5, #508; [0032], [0039]) into a rules engine adapted for providing processor executable procedures (FIG. 5, 510; [0031], [0044]). Means are provided for accessing (FIG. 6, #608; FIG. 7, #722; FIG. 8, #800; [0052]) an ontology (FIG. 1, #118; FIG. 5, #512) describing messages of the distributed computing service. The system includes means for obtaining (FIG. 8, #800; [0052]) a set of procedures from the rules engine for interacting with the distributed service based on the semantic interpretation specification (FIG. 3, #306; FIG. 5, #508, 510; [0032], [0039], [0044]), and means (FIG. 8, #800; [0052]) for forming distributed computing service messages based on the ontology for use in the set of procedures (FIG. 5, #504, #506, #514; [0044]). Means are further provided for interfacing (FIG. 8, #800; [0052]) with the distributed computing service using the set of procedures (FIG. 3, #306; FIG. 5, #508, 510; [0032], [0039], [0044]).

In another embodiment as found in claim 21, a method of interfacing with a distributed computing service is provided. The method comprises receiving a message from the distributed computing service (FIG 5, #504; [0044]; FIG. 7, #702; [0048]), and identifying a message type of the message for processing of the message (FIG. 7, #710; [0049]). The method accesses (FIG. 6, #608; FIG. 7, #722) an ontology specification (FIG. 1, #118; FIG. 5, #512) describing the message type and accesses (FIG. 6, #614; [0032], [0053]) a semantic interpretation specification (FIG. 1, #120; FIG. 5, #510) describing rules for semantically handling the messages, as specified in the ontology specification, with the distributed computing service based on the message type. The semantic interpretation specification is entered (FIG. 3, #304; FIG. 5, #508; [0032], [0039]) into a rules engine adapted for providing processor executable procedures (FIG. 5, 510; [0031], [0044]). The method obtains a set of procedures from the rules engine for interacting with the distributed service based on the semantic interpretation specification (FIG. 3, #306; FIG. 5, #508, 510; [0032], [0039], [0044]) and interfaces with the distributed computing service using the set of procedures in response to the message. The interfacing comprises forming a distributed computing service message based on the ontology specification and outputting the message (FIG. 5, #504, #506, #514; [0044]).

According to claim 25, a system comprises a first data processing arrangement configured to provide a distributed computing service (FIG. 8, #830; [0054]) and a data storage arrangement (FIG. 1, #108; FIG. 8, #824, #826; [0035], [0054]). The data storage arrangement contains an ontology specification (FIG. 1, #118; FIG. 5, #512) and a semantic interpretation specification (FIG. 1, #120; FIG. 5, #510), wherein the ontology specification describes messages of the distributed computing service, and the semantic interpretation specification describes rules for semantically handling the messages, as specified in the ontology specification, used to interface with the distributed computing service. A second data processing arrangement (FIG. 8, #801; [0052]) has a rules engine adapted for providing processor executable procedures (FIG. 5, #500, #510; FIG. 8, #838; [0031], [0044], [0053]). The second data processing arrangement configured to receive a request to interface with the distributed computing service (FIG 5, #504; [0044]; FIG. 7, #702; [0048]), access (FIG. 6, #608; FIG. 7, #722) the ontology specification (FIG. 1, #118; FIG. 5, #512) from the data storage arrangement, and access (FIG. 6, #614; [0032], [0053]) the semantic interpretation specification from the data storage arrangement. The second data processing arrangement enters (FIG. 3, #304; FIG. 5, #508; [0032], [0039]) the semantic interpretation specification into the rules engine and obtains a set of procedures from the rules engine for interacting with the distributed service based on the semantic interpretation specification (FIG. 3, #306; FIG. 5, #508, 510; [0032], [0039], [0044]). The second data processing arrangement interfaces with the distributed computing service using the set of procedures. The interfacing includes forming distributed computing service messages based on the ontology specification (FIG. 5, #504, #506, #514; [0044]).

VI. Grounds of Rejection

A. Claims 1, 2, 6-9, 12-14, 18, 19, and 25-27 stand rejected under 35 U.S.C. §102(e) as being anticipated by "Eanes" (U.S. Pub. No. 2003/0005412 to Eanes).

B. Claims 3, 4, 10, 11, 15, 16, 21-23, and 28 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Eanes in view of "Ott" (U.S. Pub. No. 2002/0150093 to Ott et al.).

VII. Argument

A. The rejection of claims 1, 2, 6-9, 12-14, 18, 19, and 25-27 should be reversed because the Examiner has not shown that all the limitations of these claims are anticipated by Eanes.

Claims 1, 2, 6-9, 12-14, 18, 19, and 25-27

The limitations of claim 1 recite accessing an ontology specification and a semantic interpretation specification. The ontology specification describes messages of the distributed computing service, and the semantic interpretation specification describes rules for semantically handling the messages specified in the ontology specification. The semantic interpretation specification e entered into a rules engine, and procedures are obtained from the rules engine for interacting with the distributed service based on the semantic interpretation specification as entered in the rules engine. Thus, the claimed process for obtaining the procedures frees a user from having to build the programmatic interfaces [0041].

The Examiner has not shown that Eanes teaches these limitations. The Examiner cited Eanes' paragraphs [0026] and [0029]-[0045] as containing elements corresponding to the entering of the semantic interpretation specification into the rules engine for obtaining a set of procedures from the rules engine for interacting with the distributed service. These paragraphs, however, describe conceptual component operations without reference to descriptions of messages in the ontology description and without reference to a rules engine for providing procedures for interacting with the distributed service based on the semantic interpretation specification. Eanes generally describes conceptual component operations that may be used to describe agent behavior [0029]. Therefore, the limitations of claim 1 are not shown to be taught by Eanes.

The Examiner responded to Appellant's arguments with: "as defined by Applicant's specification ... an ontology is a 'metadata schema that provides a formal, machine-processable, explicit specification of a set of terms ... [and] a semantic interpretation specifications [sic] 'express rules for handling the message using the published ontology' ... As such, the EBNF rules for processing a document shown by Eanes in para. [0029]-[0044], and the example given by Eanes in para. [0057]-[0058] which shows an actual XML schema

utilizing the EBNF rules to process particular messages meet the claim limitations." Appellant respectfully submits that those skilled in the art will recognize that Eanes' extended Backus-Naur Form (BNF) neither teaches nor suggests the claimed method involving a rules engine for providing the procedures.

Those skilled in the art will recognize that BNF is used to express context-free grammars that specify formal languages. In contrast, an example embodiment of the claimed rules engine is built with Java Expert System Shell (JESS), which as a rules engine is fundamentally different from a BNF specification. Thus, the claimed entry of the "semantic interpretation specification into a rules engine adapted for providing processor executable procedures" and the "obtaining a set of procedures from the rules engine for interacting with the distributed service" is neither taught nor suggested by Eanes' BNF specification of operations in a domain's procedural model ([0026]). These limitations are not anticipated by Eanes.

Independent claims 7, 13, 19, and 25 include limitations similar to those of claim 1. Claims 2, 6, 8, 9, 12, 14, 18, 26, and 27 depend from the base claims discussed above. Therefore, the Examiner has not shown that Eanes teaches all the limitations of these claims for at least the reasons set forth above.

The rejection of claims 1, 2, 5-9, 12-14, 17-20 and 25-27 should be reversed because all the limitations are not shown to be taught by Eanes.

B. The rejection of claims 3, 4, 10, 11, 15, 16, 21-23, and 28 should be reversed because the Examiner has not established a *prima facie* case of obviousness of the claims under 35 U.S.C. §103(a) over Eanes in view of "Ott" (U.S. Pub. No. 2002/0150093 by Ott et al.).

The Examiner has failed to establish a *prima facie* case of obviousness of claims 3, 4, 10, 11, 15, 16, 21-23, and 28 because all the limitations have not been shown to be suggested by the Eanes-Ott combination and a proper motivation for modifying Eanes with teachings of Ott has not been provided.

Claims 3, 4, 10, 11, 15, 16, and 28

Claim 3 depends from claim 1 and further recites that the semantic interpretation specification comprises an expert system interpretable specification. Claim 4 depends from claim 3 and specifies that the semantic interpretation specification comprises rules usable with a rule-based expert system. Thus, claim 3 together with claim 1, sets forth entering the semantic interpretation specification, which is expert system interpretable, into a rules engine adapted for providing processor executable procedures. Claims 10, 11, 15, 16, 23, and 28, include limitations similar to those of claims 3 and 4.

Ott does not suggest using an expert system interpretable semantic interpretation specification for providing processor executable procedures for interacting with the distributed service. Rather, Ott teaches a specific use of an expert system, and that specific use is not suggestive of the claimed expert system interpretable semantic interpretation specification for providing processor executable procedures for interacting with the distributed service. Ott teaches routing of data through a network based on semantics of the data (Abstract). Profiles are used to determine which output port(s) will get data received at an input port. If a profile of an output port matches the content of data received at an input port, that output port will get the received data ([0081]). Ott applies expert systems and rule-based programming to the aggregation of semantic profiles [0048-0081]. There is no apparent suggestion of the claimed use of expert system interpretable semantic interpretation specification for providing processor executable procedures for interacting with the distributed service. Thus, the Examiner has not shown that the Eanes-Ott combination suggests the limitations of claims 3, 4, 10, 11, 15, 16, 21-23, and 28.

The asserted motivation for modifying the teachings of Eanes with teachings of Ott is improper. The Office Action states that "it would have been obvious ... to modify the method of Eanes with the teachings of Ott ... by the fact that Eanes explicitly states that the process of generating agents can be automated ... based on the rules provided and Ott explicitly states that the primary purpose of an expert system can be used to automate processes normally performed by humans ..." The asserted motivation is improper because it does not suggest the specifically claimed manner in which the semantic interpretation specification, which describes rules for semantically handling the

messages specified in the ontology specification, comprises an expert system interpretable specification; the Office Action only generally cites the automation that may be achieved with expert systems and Eanes' suggestion of automation. There is no evidence presented, nor is it apparent, in what manner Eanes structure would be modified to include an expert system. Eanes teaches that an "embodiment of present invention includes a set of component metadata relating all available software components to relevant ontologies for automated combination of components into software agents." However, there is no evidence presented that suggests how or whether Eanes automated approach could be improved by an expert system. Nor is there any evidence presented that Eanes' automated approach is in any way deficient. Thus, the general desire for automation and the general capabilities of expert systems are insufficient to motivate the specifically claimed combination, and the asserted motivation is unsupported by evidence.

Claim 21, 22, and 23

The Examiner has not shown that the Eanes-Ott combination suggests that procedures are obtained from a rules engine based on a received message type. According to claim 21, the limitations include: "receiving a message from the distributed computing service; identifying a message type of the message for processing of the message; ... entering the semantic interpretation specification into a rules engine adapted for providing processor executable procedures; obtaining a set of procedures from the rules engine for interacting with the distributed service based on the semantic interpretation specification; and interfacing with the distributed computing service using the set of procedures in response to the message, wherein the interfacing comprises forming a distributed computing service message based on the ontology specification and outputting the message." Thus, procedures are obtained from a rules engine based on a received message type.

Eanes creates software agents from reusable components and does so using a metadata description of data and procedural models for a specific knowledge domain ([0015]). There is no apparent relevance of Eanes' teachings to the claimed obtaining procedures from a rules engine for interfacing with a distributed computing service based on a message type of a received message.

Eanes apparently creates the agents and procedures for interfacing with the agents prior to having received any type of message for interfacing with the agents.

Ott neither teaches nor suggests the limitations of obtaining procedures for interfacing with the distributed service based on the message type. As explained above in regards to claim 3, Ott matches a profile of an output port to the content of data received at an input port for determining whether that output port will get the received data ([0081]). Thus, there is clearly no suggestion of obtaining procedures as is claimed, and the Examiner has not shown that the Eanes-Ott combination suggests all the limitations of claim 21.

The asserted motivation for modifying Eanes with teachings of Ott is unsupported by evidence and improper. The Examiner asserted that "it would have been obvious ... to modify the method of Eanes with the teachings of Ott ... by the fact that a client must receive a message, either directly or indirectly, from a Web service describing the services offered by the service and how to further communicate with that service." It is respectfully submitted that the Examiner has not provided any evidence that indicates that Eanes' approach for receiving a message is in any manner deficient. Nor has the Examiner provided any evidence that Ott's matching of profiles to received messages would in any manner improve Eanes' creation of software agents. Therefore, the asserted motivation is unsupported by evidence and improper.

Claims 22 and 23 depend from claim 21 and a *prima facie* case of obvious has not been established for the reasons set forth above.

Appellant respectfully requests reversal of the rejections because the Examiner failed to establish a *prima facie* case of obviousness.

VIII. Conclusion

In view of the above, Appellant submits that the rejections are improper, the claimed invention is patentable, and that the rejections of claims 1-4, 6-16, 18-19, 21-23, and 25-28 should be reversed. Appellant respectfully requests reversal of the rejections as applied to the appealed claims and allowance of the entire application.

Respectfully submitted,

CRAWFORD MAUNU PLLC
1270 Northland Drive, Suite 390
Saint Paul, MN 55120

By: _____
Name: LeRoy D. Maunu
Reg. No.: 35,274

APPENDIX OF APPEALED CLAIMS FOR APPLICATION NO. 10/695,461

1. A processor-implemented method for interfacing with a distributed computing service, comprising:

accessing an ontology specification describing messages of the distributed computing service;

accessing a semantic interpretation specification that describes rules for semantically handling the messages, as specified in the ontology specification, with the distributed computing service;

entering the semantic interpretation specification into a rules engine adapted for providing processor executable procedures;

obtaining a set of procedures from the rules engine for interacting with the distributed service based on the semantic interpretation specification;

receiving a request for interfacing with the distributed service; and

interfacing with the distributed computing service using the set of procedures in response to the request, wherein the interfacing comprises forming distributed computing service messages based on the ontology specification.

2. The method of claim 1, wherein the distributed computing service comprises a Web service.

3. The method of claim 1, wherein the semantic interpretation specification comprises an expert system interpretable specification.

4. The method of claim 3, wherein the semantic interpretation specification comprises rules usable with a rule-based expert system.

6. The method of claim 1, wherein interfacing with the distributed computing service using the set of procedures comprises forming a service bridge having a generic programmatic interface adapted to receive the request.

7. An apparatus, comprising:

a data transfer interface for providing data connections to a distributed computing service; and

a processor arranged to :

access an ontology specification describing messages of the distributed computing service;

access a semantic interpretation specification that describes rules for semantically handling the messages, as specified in the ontology specification, used to interface with the distributed computing service;

enter the semantic interpretation specification into a rules engine adapted for providing processor executable procedures;

obtain a set of procedures from the rules engine for interacting with the data transfer service based on the semantic interpretation specification; and

interface with the distributed computing service via the data transfer interface using the set of procedures, wherein the interfacing includes forming distributed computing service messages based on the ontology specification.

8. The apparatus of claim 7, wherein the data transfer interface comprises a network interface.

9. The apparatus of claim 8, wherein the distributed computing service comprises a Web service.

10. The apparatus of claim 8, wherein the semantic interpretation specification comprises an expert system interpretable specification.

11. The apparatus of claim 10, wherein the expert system rules comprise rules usable with a rule-based expert system.

12. The apparatus of claim 7, further comprising a memory and a service bridge module stored in the memory, the service bridge module operable via the processor to activate the set of procedures based on instructions from a generic programmatic interface of the service bridge module.

13. A computer-readable medium configured with instructions for causing a processor to perform steps comprising:

accessing an ontology specification describing messages of the distributed computing service;

accessing a semantic interpretation specification describing rules for semantically handling the messages, as specified in the ontology specification, with a distributed computing service;

entering the semantic interpretation specification into a rules engine adapted for providing processor executable procedures;

obtaining a set of procedures from the rules engine for interacting with the distributed service based on the semantic interpretation specification;

receiving a request for interfacing with the distributed service; and

interfacing with the distributed computing service using the set of procedures in response to the request, wherein the interfacing comprises forming distributed computing service messages based on the ontology specification.

14. The computer-readable medium of claim 13, wherein the distributed computing service comprises a Web service.

15. The computer-readable medium of claim 13, wherein the semantic interpretation specification comprises an expert system interpretable specification.

16. The computer-readable medium of claim 15, wherein the semantic interpretation specification comprises rules usable by a rule-based expert system.

18. The computer-readable medium of claim 13, wherein interfacing with the distributed computing service using the set of procedures comprises forming a service bridge module having a generic programmatic interface usable to execute the set of procedures.

19. A system comprising:

means for providing a distributed computing service;

means for storing an ontology specification describing messages of the distributed computing service;

means for storing a semantic interpretation specification that describes rules for semantically handling the messages, as specified in the ontology specification, used to interface with the distributed computing service;

means for accessing the semantic interpretation specification for entry into a rules engine adapted for providing processor executable procedures;

means for accessing an ontology describing messages of the distributed computing service;

means for obtaining a set of procedures from the rules engine for interacting with the distributed service based on the semantic interpretation specification; and

means for forming distributed computing service messages based on the ontology for use in the set of procedures; and

means for interfacing with the distributed computing service using the set of procedures.

21. A method of interfacing with a distributed computing service comprising:

receiving a message from the distributed computing service;

identifying a message type of the message for processing of the message;

accessing an ontology specification describing the message type;

accessing a semantic interpretation specification describing rules for semantically handling the messages, as specified in the ontology specification, with the distributed computing service based on the message type;

entering the semantic interpretation specification into a rules engine adapted for providing processor executable procedures;

obtaining a set of procedures from the rules engine for interacting with the distributed service based on the semantic interpretation specification; and

interfacing with the distributed computing service using the set of procedures in response to the message, wherein the interfacing comprises forming a distributed computing service message based on the ontology specification and outputting the message.

22. The method of claim 21, wherein the distributed computing service comprises a Web service.

23. The method of claim 21, wherein the semantic interpretation specification comprises expert system rules.

25. A system comprising:

a first data processing arrangement configured to provide a distributed computing service;

a data storage arrangement containing an ontology specification and a semantic interpretation specification, wherein the ontology specification describes messages of the distributed computing service, and the semantic interpretation specification describes rules for semantically handling the messages, as specified in the ontology specification, used to interface with the distributed computing service;

a second data processing arrangement having a rules engine adapted for providing processor executable procedures, the second data processing arrangement configured to:

receive a request to interface with the distributed computing service;

accessing the ontology specification from the data storage arrangement;

access the semantic interpretation specification from the data storage arrangement;

enter the semantic interpretation specification into the rules engine;

obtain a set of procedures from the rules engine for interacting with the distributed service based on the semantic interpretation specification; and

interface with the distributed computing service using the set of procedures, wherein the interfacing includes forming distributed computing service messages based on the ontology specification.

26. The system of claim 25, wherein the distributed computing service comprises a Web service.

27. The system of claim 25, wherein the a data storage arrangement is adapted for providing the semantic interpretation specification via a network.

28. The system of claim 25, wherein the semantic interpretation specification comprises expert system rules.

APPENDIX OF EVIDENCE FOR APPLICATION NO. 10/695,461

Appellant is unaware of any evidence submitted in this application pursuant to 37 C.F.R. §§ 1.130, 1.131, and 1.132.

APPENDIX OF RELATED PROCEEDINGS FOR APPLICATION NO. 10/695,461

Appellant is unaware of any related appeals, interferences or judicial proceedings.